Scheduling strategies for multicast transmissions: Why ? How ?

Ecole d ’été des Réseaux Hauts Débits et Multimédia (RHDM ’99), Bretagne, France, septembre 1999.

vincent.roca@lip6.fr - http://www-rp.lip6.fr/~roca

Goals of this presentation

- **Multicast is not only a routing (level 3) task**
- **Look at some higher level issues…**
- **Very large problem-space ⇒ we only consider scheduling techniques...**

Outline

- 1- the context
- 2- introduction to layered scheduling
- 3- scheduling schemes for data-flows known in advance
- 4- scheduling schemes for data-flows generated on-the-fly
- 5- a library for the experimentation of multicast algorithms

1- The context

Classification of applications from a data-flow point of view

- Does the receiving application need all the data or is a subset sufficient?

  Example: file transfer versus multi-definition video

- Is data known in advance or generated on-the-fly ?

  Example: binary distribution versus audio-conference

- Is the data stream continuous or intermittent ?

  Example: video versus white-board

- Are asynchronous starts (also called late-arrival) possible or not ?

  Example: video-on-demand

The problem space of multicast transmissions

Multicast = efficient transmission of data to a group of receivers

- "efficient…"
  - **avoid (limit?) packet duplication**
  - **use *congestion control* schemes (e.g. for TCP friendliness)**
  - **offer a good *scalability***
- "data…"
  - **use fully / partial / no reliability ?**
  - **what data to deliver ?**
  - **handle late arrivals ?**
- "group of receivers..."
  - **take receiver *heterogeneity* into account ?**

    (network / processing power / user desires)

Scheduling strategy goals

- Applications have widely different needs
  - ⇒ **no single multicast solution**
- Scheduling can help to do
  - **congestion control**
  - **late arrival handling**
  - **heterogeneity support**
- But they must usually be associated to other techniques...

2- Introduction to layered scheduling schemes

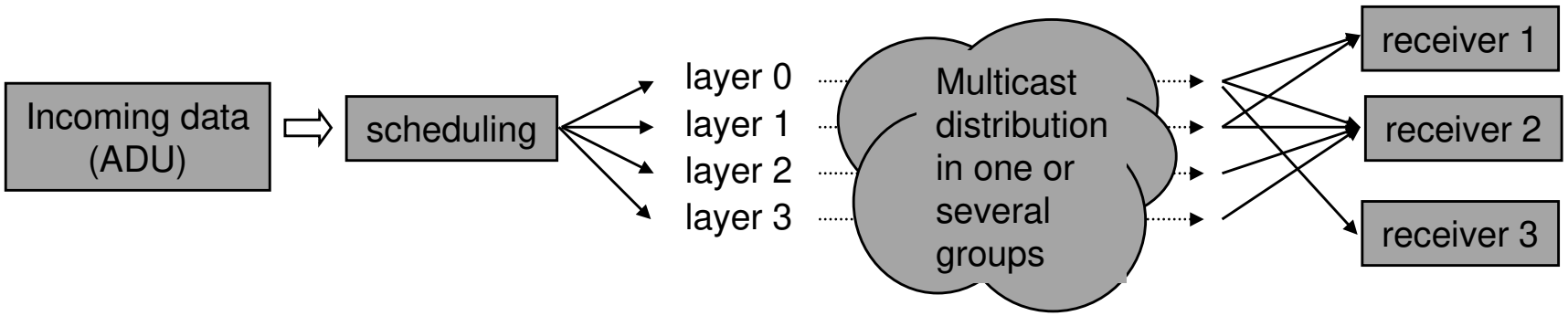
- The class of solutions we are interested in...
- Principle

> **Split Application Data Units (ADU) into Data Units (DU) and schedule them appropriately in the available transmission layers**

- The notions of ADUs and DUs
  - **an ADU can be a complete file, a block of a file, an image, etc.**
  - **an ADU may or may not be processed out-of-order**
  - **a DU is often a datagram ready to be sent, otherwise it is a data block**

Layered scheduling… cont'

- The notion of layers
  - **several transmission layers (or "channels" or "levels")**
  - **can be *cumulative*** **subscribe to layers {0; 1; … i}**
    or *not* **subscribe only to layer i**
  - **usually "a layer" ⇔ "a multicast group" …**
    but not compulsory!
  - **transmissions occur at a *fixed rate* on each layer**
  - **a receiver decides what layer to receive and when**



- must be associated to other techniques...

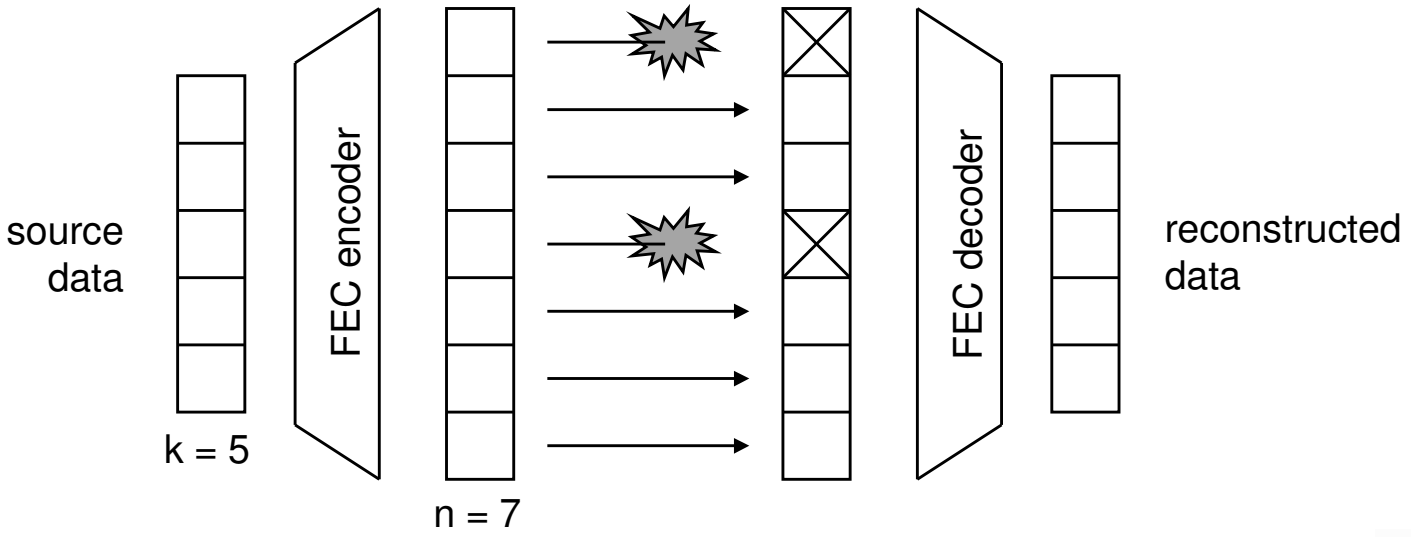
Associated techniques: reliability

- Use FEC (Forward Error Correction) [Rizzo97]
  - **Sender: uses FEC (k, n)**

    for k original data packets, add n-k FEC encoded redundant packets

    ⇒ total of n packets sent
  - **Receiver:**

    as soon as it receives *any* k packets out of the n, it reconstructs the original k packets



Reliability… cont'

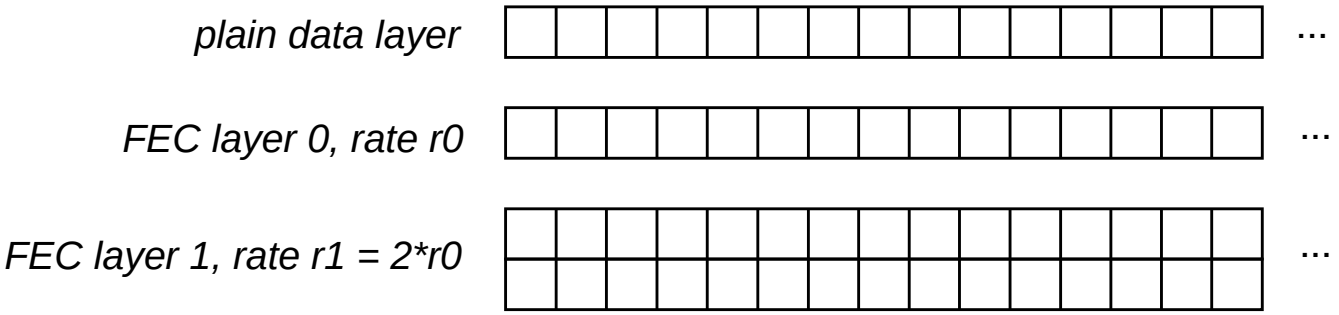
- **Limitations (case of Reed-Solomon):**
  - practical limitation to k (~32 max)

    ⇒ split original data into blocks of k packets each
  - decoding complexity increases with the number of FEC packets received

- How to use FEC
  - **Solution 1: transmit FEC packets along with plain data**
    - limits: everybody receives the same FEC packets!
  - **Solution 2: use layered FEC**
    - a receiver chooses how many FEC packets to receive, and
    - how fast to receive them

Reliability… cont'

- **Example: 2 FEC layers, 1 plain data layer**



- **Partial (or total) reliability according to how it is used**

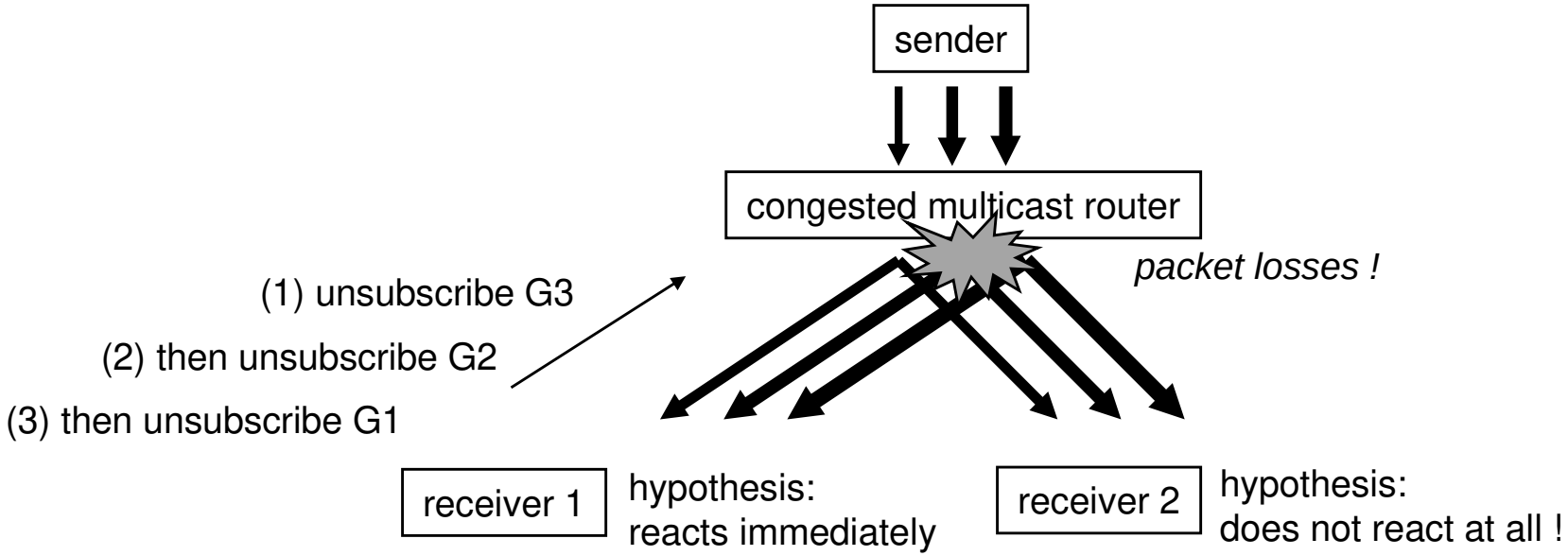
- use or ARQ (Automatic Repeat reQuest, i.e. ACK/NAK protocols)
  - **example: SRM**

- use mixed ARQ-FEC

Associated techniques: congestion control

- Congestion control for layered scheduling
  - **receiver driven adaptation [McCanne96]**

    according to the experienced loss rate a receiver chooses to (un)subscribe to one or more multicast groups
  - **but there are problems...**



Congestion control… cont'

- **add synchronization points / probes [Vicisano98]**
  - (un)subscription is only possible at a synchronization point
    ⇒ multicast tree rapidly updated
  - the sender artificially increases the transmission rate to simulate the consequences of subscribing to an additional group
- **open issue: fast responses in front of congestion?**

- A new approach: use filtering [Luby99]
  - **perform packet filtering within routers**
  - **must be widely deployed to be efficient...**

3- Scheduling schemes for data flows known in advance

- many target applications: file transfer, video-on-demand, software distribution…
- requires a *prior knowledge of data* (single ADU)
- usually all the receivers need to *receive everything*

Solution 1: MCM (Multiple Channel Multicast)

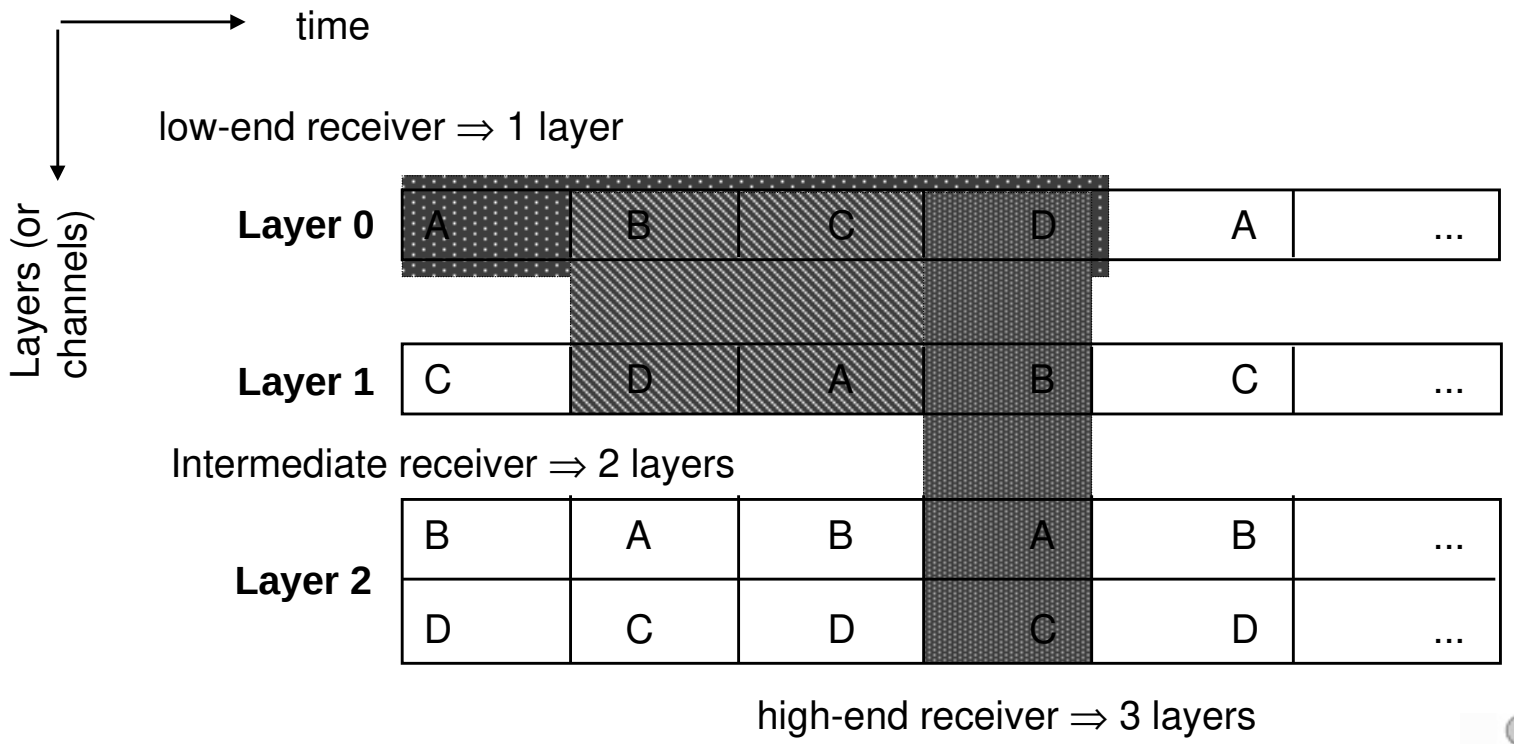
- [Vicisano98][Donahoo99]

- Features:
  - **cumulative scheduling (layer i takes advantage of layers {0, … i-1})**
  - **supports late-arrivals**
  - **supports heterogeneity**

- Gathers several solutions…
- Common principles
  - data partitioned into n_c equal parts
  - use several channels (or layers)
  - cyclic transmissions in each channel
  - a receiver can join one or more channels at a time (according to its available bandwidth)
- Differ on the mapping of parts to channels

Solution 1… cont’

- First channel scheduling
  - **each part is sent on each channel**
  - **Example: 4 parts, 3 layers, no FEC**

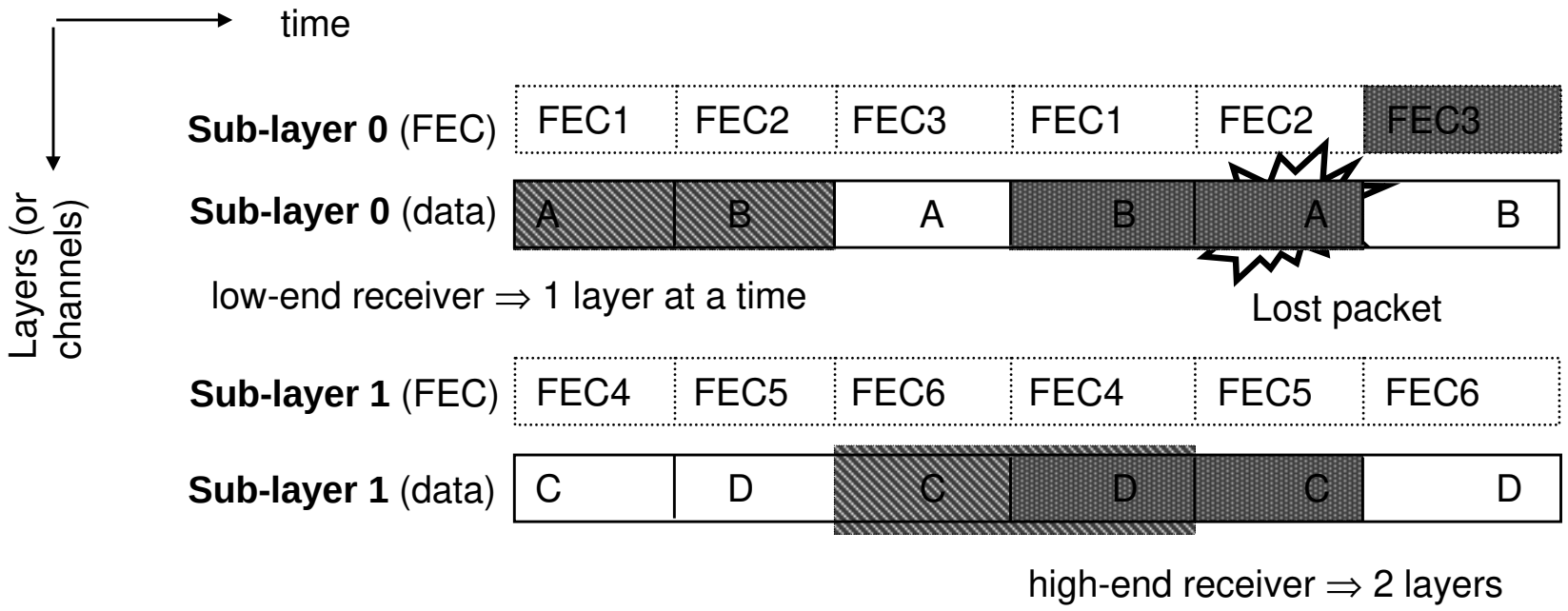


Solution 1… cont'

- The PO (Partition Organization) channel scheduling [Donahoo99]
  - **Principles:**
    - data partitioned into n_c equal parts
    - each part is allocated to a single channel but a channel can handle several parts
    - each channel is composed of:
      - a sub-channel for plain data, and
      - a sub-channel for FEC data
    - cyclic transmissions in each channels
    - a receiver can join any channel at a time (according to the available bandwidth)
    - to receive everything, a receiver must have joined all the channels at some time

Solution 1… cont’

- **Example: 4 parts (A, B, C, D), 2 layers, FEC**



Solution 1… cont'

- Benefits:
  - **separation original/FEC packets is efficient (use FEC only if required)**
  - **simple scheduling algorithm**
  - **more efficient than the SO (Session Organization) scheduling scheme proposed by [Vicisano97]**
  - **good behavior in case of uncorrelated packet losses**

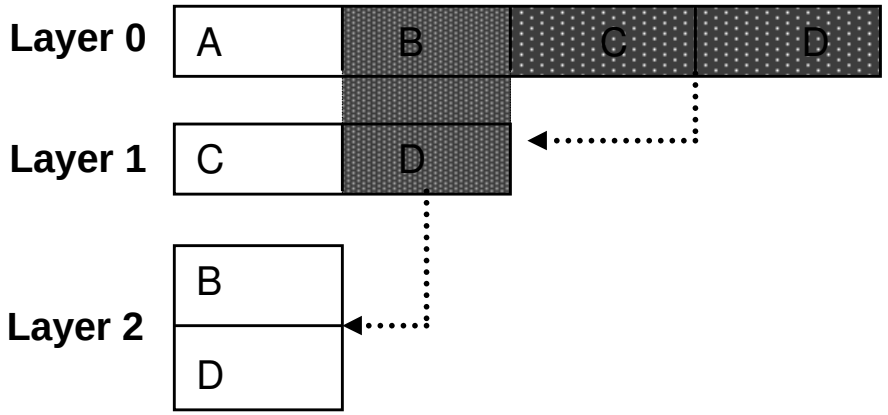
- But…
  - **continuous burst losses may create problems**
  - **limited heterogeneity (supports only an additional distribution of tx rates)**

Solution 2 : MMG (Multiple Multicast Groups)

- [Bhattacharyya98]
- The main interest of the paper is the *mathematical analysis* !
- Features
  - **cumulative scheduling**
  - **supports heterogeneity**
  - **supports any scale of transmission rates for each layer**
  - **but does NOT support late-arrivals**
- Principles
  - data partitioned into n_c equal parts
  - use several channels (or layers)
  - a receiver joins one or more channels
  - a receiver unsubscribes as soon as all the parts are received

Solution 2… cont'

- Example: 4 parts (A, B, C, D), 3 layers, no FEC, power of 2 transmission scale



- But:
  - **an optimal scheduling requires the complete knowledge of {receivers; reception rates}**

Solution 3: skyscraper broadcasting

- [Hua97] [Eager98]

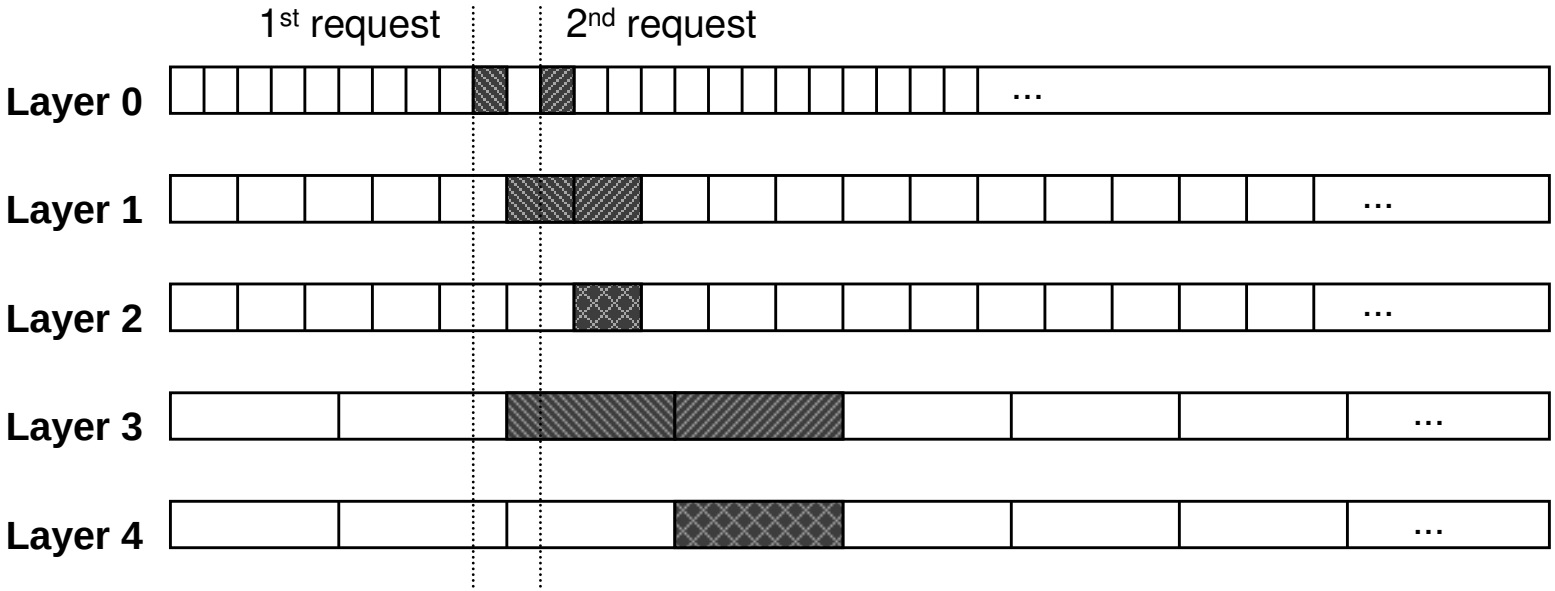
- Features
  - **quasi instantaneous asynchronous starts**
  - **but assumes no heterogeneity among receivers !**
  - **more suited to video-on-demand servers**

- Principles
  - **consider a file [Hua97] (or n files [Eager98])**
  - **K layers are used**
  - **a receiver subscribes to 2 layers at a time, then switch**
  - **file is first segmented into K fragments of increasing size**

    progression: {1, 2, 2, 5, 5, 12, 12, 25, 25, 52, 52, 105, 105…}
  - **send continuously the same fragments on each layer**

Solution 3… cont’

- Example: 5 layers, 2 receivers starting at different times

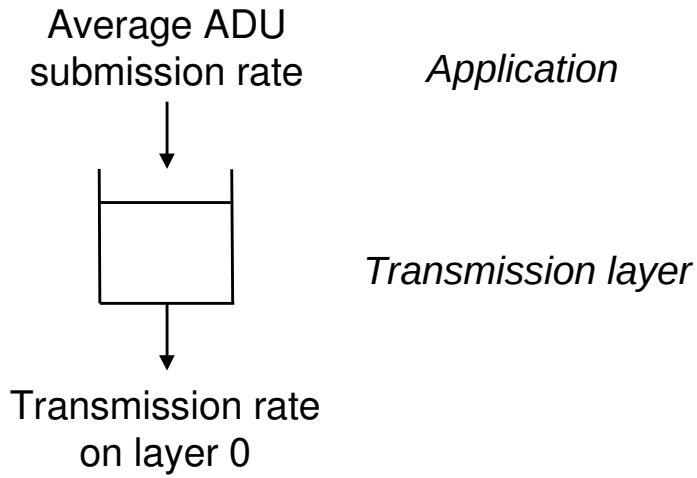


4- Scheduling schemes for data flows generated on-the-fly

- target applications: cooperative work applications (audio/video-conference, white board, application sharing environments, etc.)
- data is *generated in real-time*
- receivers need either to *receive everything or just a subset*
- management of late-arrivals done at application level if required

Consequences...

- Data submitted at any time as a collection of Application Data Units (ADUs)
- Consequences:
  - **1- complicates scheduling**
  - **2- can create packet duplications**
  - **3- the average ADU submission rate must be lower than the basic transmission rate... otherwise the application is blocked**



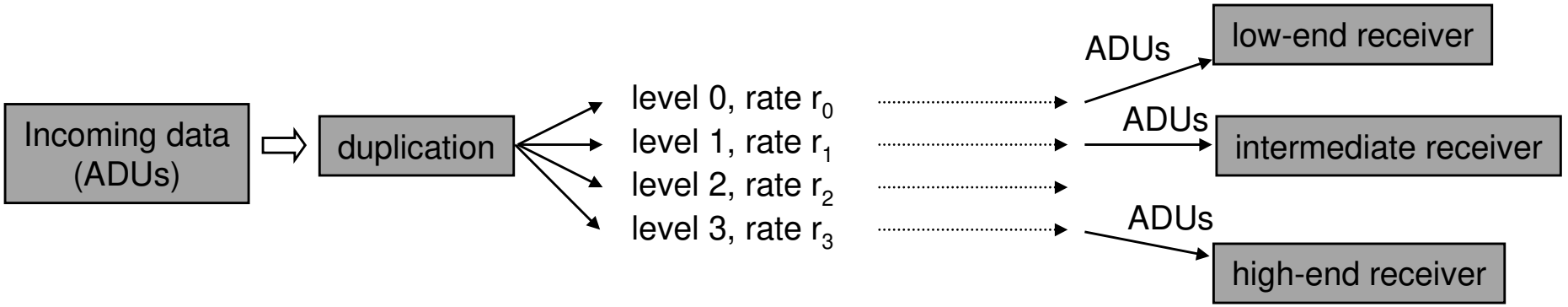
- These problems occur with the DSS and GMMG algorithms

Solution 1: hierarchical coding

- [Bolot94]
- Features:
  - **based on a multi-definition decomposition of data**
  - **one layer per definition**
  - **cumulative scheduling**
  - **supports heterogeneity**
  - **very popular**
- But...
  - **receivers don’t receive the same data**
  - **limited field of application (video-conference) !**

Solution 2: Destination Set Splitting (DSS)

- [Ammar92]
- Features
  - **use several levels, each one with a different transmission rate**
  - **straightforward**
  - **used with bursty data flow**



- Not so bad but...
  - **this is not a cumulative scheduling**
  - **high network load close to the sender (traffic is concentrated)**

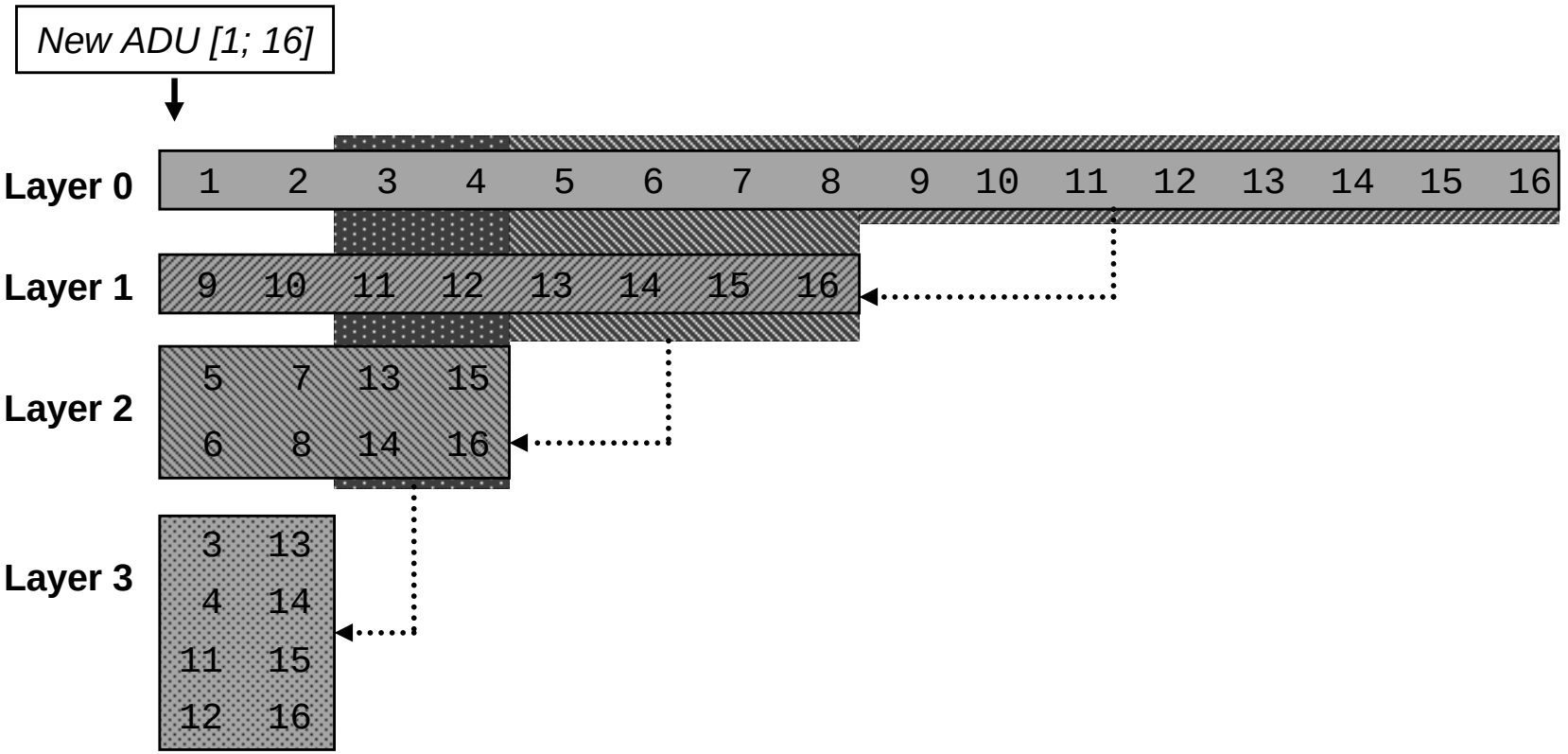
Solution 3: Generalized MMG (GMMG)

- [Roca99][Roca99b]

- Features
  - **use several layers, each one with a different transmission rate**
  - **cumulative scheduling**
  - **used with bursty data flow**

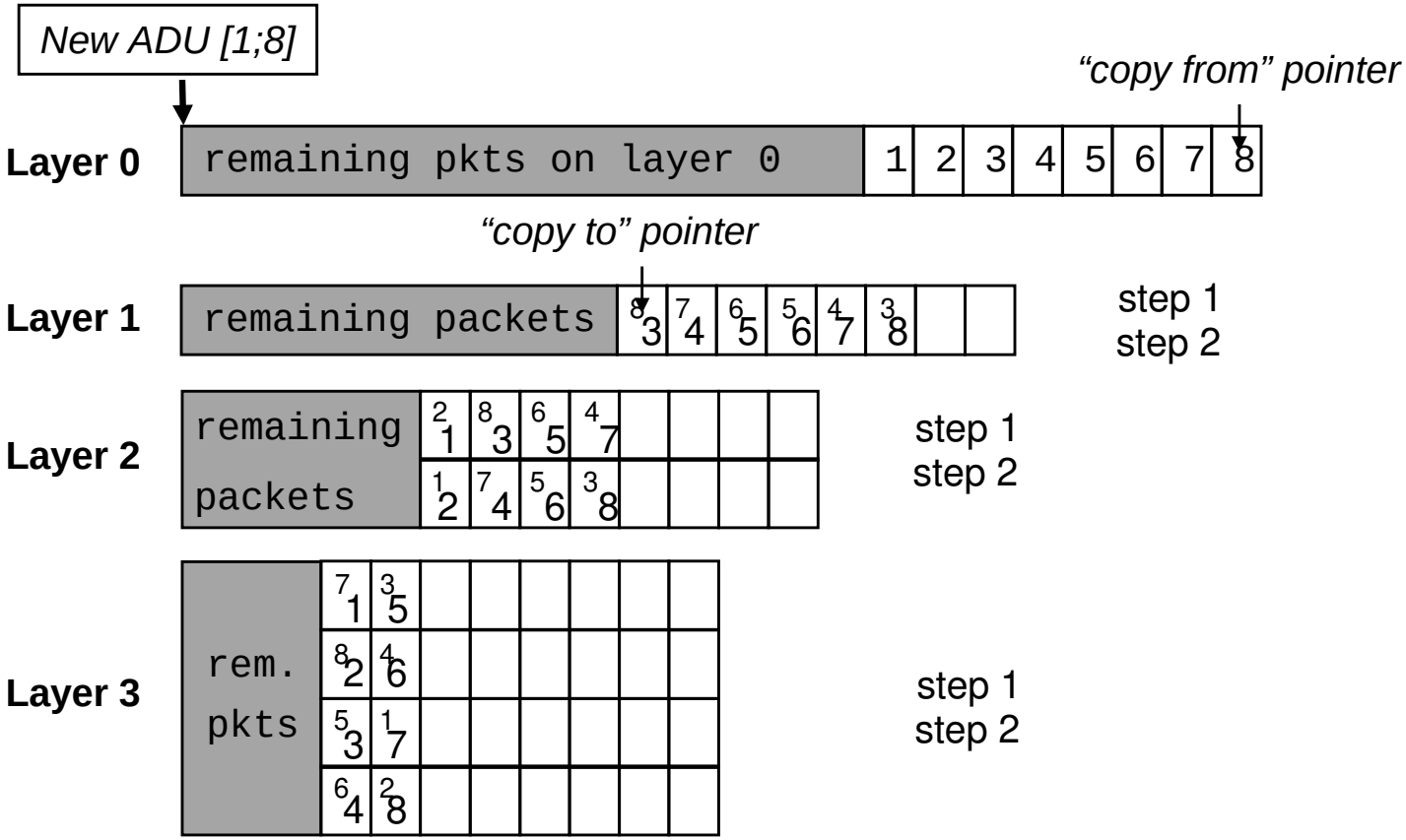
Solution 3: the case of an isolated ADU

- Similar to MMG in case of an isolated ADU
- Example: ADU 1 ⇒ 16 packets



Solution 3: the general case

- Some packets may remain, waiting to be transmitted
- The algorithm consists in filling empty cells
- 2 steps: copy, then reorder



Solution 3: bilan

- This algorithm is:
  - **simple**
  - **efficient**
  - **accommodates any scale of transmission rates (not only power of 2)**
  - **reduced packet duplication at the sender**

- But…
  - **duplicated packets at the receiver**
  - **performances depend on two parameters:**
    - average ADU submission rate versus layer 0 transmission rate
    - ADU size

5- A library for the experimentation of multicast algorithms

- [Roca99b]

- Experiment various algorithms for multicast transmissions
- Follows the building block approach of the RMRG [Whetten99]

- Quite simple upper API (Application Programming Interface)

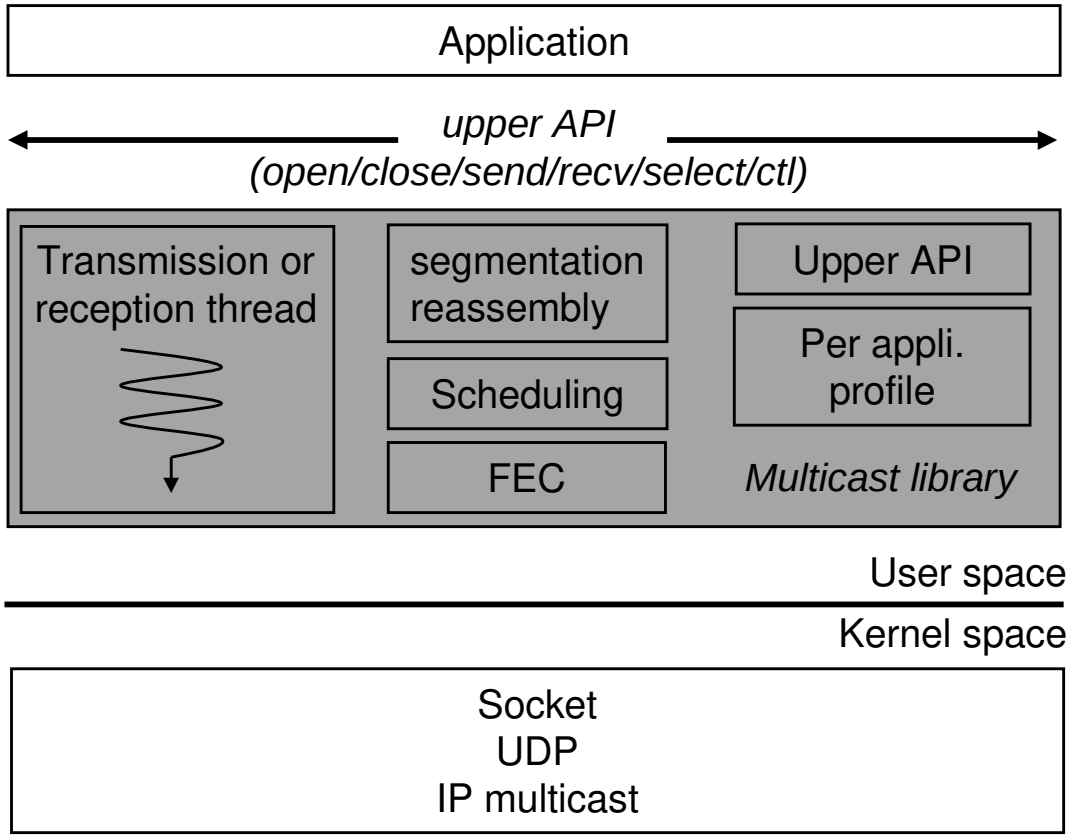
```
open / close / send / recv / select / ctl
```

  - **Simplifies the migration of existing applications**

- General profile determined at compilation time
- Main parameters can be modified dynamically

- Freely available (source/documentation)

Architecture of the library



Supported functionalities (version 0.9b, July 1999)

- Scheduling schemes
  - **DSS, GMMG and MMG**
  - **supports both data generated on-the-fly and known in advance**
- Reliability
  - **layered FEC (optional)**
- Applications
  - **two traffic generator applications**
- still under development…
- lack in particular congestion control

6- Conclusions

- Layered schemes are a key class of solutions for the handling of heterogeneous receivers
- Two general cases:
  - **data known in advance, and**
  - **data generated on-the-fly**
- Scheduling schemes are still the subject of many research…
- This is all the more true for the associated functionalities (congestion control, fairness, reliability, etc.)
- Our library enables true-size experiments…

Selected bibliography

- Our approach

[Roca99] V. Roca, “Packet scheduling for heterogeneous multicast transmissions”, Protocols for High Speed Networks (PfHSN’99), August 1999.

[Roca99b] V. Roca, “Design of a library for the heterogeneous multicast distribution of data flows generated on the fly”,work in progress, July 1999.

- Layered scheduling

[Donahoo99] M. Donahoo, M. Ammar, E. Zegura, “Multiple-channel multicast scheduling of scalable bulk-data transport”, INFOCOM’99, March 1999.
NB: an extended version is available on the authors’s home page

[Clerget98] A. Clerget, W. Dabbous, “Organizing data for reliable multicast over satellite links”, 5th Conference on Computer Communications (Africom’98), 1998.

[Ammar92] M. Ammar, L. Wu, “Improving the throughput of point-to-multipoint ARQ protocols through destination set splitting”, INFOCOM’92, May 1992.

[Bhattacharyya98] S. Bhattacharyya, J. Kurose, D. Towsley, R. Nagarajan, “Efficient multicast flow control using multiple multicast groups”, INFOCOM’98, 1998.

[Viscisano98] L. Vicisano, “Notes on a cumulative layered organisation of data packets across multiple streams with different rates”, Research notes RN/98/25, work in progress, 1998.

Selected bibliography… cont'

- Skyscraper scheduling

[Hua97] K. Hua, S. Sheu, "Skyscraper broadcasting: a new broadcasting scheme for metropolitan video-on-demand systems", ACM SIGCOMM'97, September 1997.

[Eager98] D. Eager, M. Vernon, "Dynamic broadcasts for video-on-demand", 4th workshop on multimedia information systems (MIS'98), Turkey, September 1998.

- FEC

[Rizzo97] L.. Rizzo, "Effective erasure codes for reliable computer communication protocols", ACM Computer Commincation Reviez, V. 27, N. 2, April 1997.

[Byers98] J. Byers, M. Luby, M. Mitzenmacher, A. Rege, "A digital fountain approach to reliable distribution of bulk data", TR-98-013, also in SIGCOMM'98, 1998.

[Nonnenmacher97] J. Nonnenmacher, E. Biersack, D. Towsley, "Parity-based loss recovery for reliable multicast transmissions", ACM SIGCOMM'97, September 1997.

Selected bibliography… cont'

- Congestion control for layered scheduling

[McCanne96] S. McCanne, V. Jacobson, M. Vetterli, "Receiver-driven layered multicast", ACM SIGCOMM'96, October 1996.

[Vicisano98] L.. Vicisano, L. Rizzo, J. Crowcroft, "TCP-like congestion control for layered multicast data transfer", IEEE INFOCOM'98, February 1998.

- Other congestion control techniques

[Luby99] M. Luby, L. Vicisano, T. Speakman, "Heterogeneous multicast congestion control based on router packet filtering", work in progress, presented at the RMRG meeting, Pisa, June 1999.

[Clerget99] A. Clerget, W. Dabbous, "TUF: a tag-based UDP multicast flow control protocol", INRIA Technical note, July 1999.

Selected bibliography… cont'

- General

[Handley99] M. Handley, B. Whetten, R. Kermode, S. Floyd, L. Vicisano, "The reliable multicast design space for bulk data transfer", Internet draft <draft-ietf-rmt-design-space-00>, June 1999.

[Quinn99] B. Quinn, K. Almeroth, "IP multicast applications: challenges and solutions", Internet draft <draft-ietf-mboned-mcast-apps-00.txt>, February 1999.

[Whetten99] B. Whetten, L. Vicisano, R. Kermode, M. Handley, S. Floyd, "Reliable multicast transport building blocks for one-to-many bulk-data transfer" Internet draft <draft-ietf-rmt-buildingblocks-00.txt>, June 1999.